CS 20: Scene Switching (17 marks)

Outcomes include:

CS20-FP3 Construct and utilize functions to create reusable pieces of code.
(a) Explore the benefits of using built-in and user-defined functions.

CS20-CP2 Use common coding techniques to enhance code elegance and troubleshoot errors throughout Computer Science 20.
(a) Discuss and implement appropriate coding style (e.g., indentation and comments) and naming conventions for the programming language used in Computer Science 20.
(d) Create internal documentation (e.g., inline comments and header comments) for a program.
(e) Discuss different types of errors (e.g., syntax, semantic and runtime) and their impacts on program execution.
(f) Identify and correct errors in a program.

	Expert - 4	**Meeting** - 3	**Progressing** - 2	**Beginning** - 1	**Incomplete** - 0
Execution	The program runs properly with no errors At least five different shapes Complex scene that changes when any key is pressed The picture reverts to the original picture when the key is released	The program runs properly with no errors At least five different shapes Scene that changes when any key is pressed The picture reverts to the original picture when the key is released	Minor errors in the code that do not prevent the code from running At least five different shapes Code may start with a blank screen. A scene that changes when any key is pressed The picture may not revert to the original	Errors in code prevent it from running	Submitted code is incomplete

			picture when the key is released		
Functions and Parameters	Functions are used for all code. The functions add to the readability of the code. All functions use parameters. Each shape is made with an original function with a meaningful name All functions are called and work properly.	At least one function with parameters is used. The function adds to the readability of the code. All functions are called and work properly.	At least one function is used but parameters are not used properly (i.e. code bypasses the parameters). Functions work properly.	No parameters were used for the functions.	No functions were used
Formatting		Code is organized and easy to read All code is formatted according to PEP8 All comments are properly formatted	Code may be difficult to follow All code is formatted according to PEP8 Not all comments are properly formatted	Code may be difficult to follow A few minor formatting errors according to PEP8 Comments are not formatted properly	Code is difficult to follow Multiple formatting errors according to PEP8

Documentation		Program includes a header with all necessary information File name relates to the program Comments for at least blocks of code that belong together Comments for all functions All variables are given meaningful names and are defined with comments		The header may be missing important information File name may not relate to the program Coding does not have enough comments Variables may not have meaningful names, or are not clearly defined.	No header File name does not relate to the program Comments are missing Variables do not have meaningful names, or are not defined.